

## DYNAMIC GRAPHIC USER INTERFACE

### RELATED APPLICATION DATA

5 This application claims the benefit of U.S. Provisional Application Serial No. 60/153,993 filed September 15, 1999, the disclosure of which is incorporated herein by reference.

### BACKGROUND OF THE INVENTION

The invention described and claimed herein relates generally to user 10 interfaces for computer systems and more particularly, graphical user interfaces. A graphical user interface or "GUI" facilitates communications between a computer operating system and a computer user. The computer system includes software and hardware, including hardware for a GUI which typically includes both a visual display (commonly a computer monitor) and selector 15 device (commonly a mouse, trackball or keyboard). Through the visual display, the computer system can deliver graphical and textual output to the user.

A conventional GUI is made up of elements presented on the visual display which allow a user to activate an application in a computer operating system. Examples of such elements are the buttons, menus, menu items, scroll 20 bars and the text input fields. Elements give the application the opportunity to

- 2 -

display information, such as zones of text or graphical displays. The layout of all the elements of a conventional GUI is generally already setup by the designer of the application, such as in the well-known WINDOWS® operating system made by Microsoft Corporation of Redmond, Washington.

5       Therefore, limited opportunities are presently available to reconfigure conventional computer applications. These typically include adjusting the position of certain controls like the toolbars. Such interfaces can be regarded as static because they do not allow the possibility of configuration by end-users. Moreover, all these elements are often of a rectangular or square shape, which  
10      is neither visually stimulating nor very pleasing to the eye.

#### SUMMARY OF THE INVENTION

The present invention provides a method and system by which users can completely reorganize a GUI to his or her personal taste. For clarity, the term element as used herein is the object which a user can move on the display  
15      device or computer screen using his or her selector device. The element is comprised of a core and an adjacent dynamic zone or edge. The core is the central part of the element. It is this core that is managed by the application program that uses the interface. The edge is the dynamic zone that preferably

surrounds the core. The edge manager is the software taking care of the management, the calculation and the display of the dynamic edges.

In a preferred embodiment, each element of the user interface in the present invention is surrounded by a dynamic edge instead of being surrounded by a static edge as in conventional systems. The present invention does not limit the shape of the core of the element to be rectangular, but allows the elements to be of any shape and in one or more pieces. The dynamic aspect of the present invention comes from the possibility of sticking or adhering together several cores in a visually coherent unit and doing it in real time.

For example, in one aspect, a user can manipulate his or her mouse and move an element, say element 1 which is green in color, towards element 2 which is red in color. The edges of element 1 merge with the edges of element 2 in real time and are updated at each movement by the edge manager, provided that the cores of the elements are not overlapping. In one aspect, this merging can be reversed and the user needs only to move one of the two elements away from the other element so that they no longer contact each other and the edges of the respective elements return to their initial size, shape and/or color.

Thus, it is an object of the present invention to provide a dynamic graphic user interface. Other and further objects will appear to those skilled in the art from the specification and drawings.

#### **BRIEF DESCRIPTION OF THE DRAWINGS**

5       Figure 1 is an illustration of two elements of the inventive graphical user interface and the two elements combined or fused together.

Figure 2 illustrates a group of elements in one embodiment of the present invention.

10      Figure 3 illustrates the group of elements in a tree arrangement illustration of an embodiment of the present invention.

Figure 4 shows an embodiment with a static element.

Figure 5a shows one embodiment of an element.

Figure 5b is a graphical illustration of the height values of the element from Figure 5a taken along line 5b - 5b.

15      Figure 6 shows an embodiment showing the fusion of edges of dynamic zones of two elements.

Figure 7 is a flow diagram summarizing an initialization procedure for one embodiment of the invention.

Figure 8 is a flow diagram illustrating a calculation process for the merging of the dynamic zones or edges of one or more elements in an embodiment.

Figure 9 is an illustration showing one use of the present invention with elements of different shapes.

5 Figure 10 is an illustration showing another use of the present invention with the elements arranged to form a design in the shape of an alligator.

Figure 11 is a further illustration showing the elements of the present invention arranged to form a design in the shape of a locomotive.

#### **DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS**

10 In the specific embodiments to be described, the invention provides a dynamic graphic user interface in an application program.

Turning now to Figures 5a and 5b, the element 10 is shown with the core 12 and edge or dynamic zone 14 in a preferred embodiment. In Figure 5b, each point of the element 10 can be assigned a numerical value. This is illustrated 15 with a conventional x - y axis in Figure 5b, with the y-axis values representing the numerical value assigned to the point, also called a height value. For all the points which are inside the core 12, the height is preferably equal to a value of one as illustrated in Figure 5b.

- 6 -

For all points which correspond to the edge 14, the height varies from a value of one, if the point is close to the core 12, and tends towards zero value as one moves away from the core 12 as also shown in Figure 5b. Persons skilled in the art will recognize that the two-dimensional representation in Figure 5b applies to other transverse sections through the element 10 and that the core 12 and edge 14 can be represented by a three dimensional array of height values.

As shown schematically in Figure 7, the array of height values is initialized at the creation of an element 10 according to the shape of the core 12. The initialization procedure is a process that associates the core 12 of an element 10 with a edge or dynamic border 14. At step 700, the graphical resources are searched and resources for the core mask 700a and core pictures 700b is searched to find one that matched the shape of the core 12. The edge of dynamic border 14 is computed using a previously defined mask of the core 12 provided for the element 10. From this mask, the shape of the core 12 is extracted and using this shape the array of height values for the edge 14 are calculated as described in connection with Figures 5a and 5b and the result is shown at step 704 in Figure 7. As can be appreciated by skilled persons, once

this array is calculated it can then be modified to render different textures and colors for each element 10.

In a preferred embodiment shown in Figure 6, a user can fuse together two or more edges 14 and 14a of two or more elements 10 and 10a. A preferred 5 operation of the edge manager of the specific embodiment for fusing two or more edges 14 and 14a is described in connection with Figure 8. At the start of Figure 8, the user moves first element (designated A) in step 800. Block 800A is the process of retrieving the list of elements that are touching element A. In detail, step 802 reviews the global list of elements and a determination is made at step 10 804 as to whether the next element is touching element A. If the answer is yes, then step 806 adds that element to the touching list and then transfers the operation to step 808. If the answer is no, the operation is transferred to step 808 which asks if the current element is the last one in the global list. If the answer is no, then the operation is transferred back to step 802 to retrieve the 15 next element in the global list, repeating the above steps for the next element. If the answer is yes, then the operation is transferred to Block 800B which is the merging block.

In detail, Block 800B retrieves the next element from the previously assembled touching list at step 810 and then, at step 812, asks if that element includes a dynamic border.

- If the answer is yes, then the border of that element is merged with the
- 5 border of element A in step 814 and operation is transferred to step 816. If the answer at step 812 is no, the operation is transferred to step 816 which asks if the current element is the last one in the touching list. If the answer is no, operation is transferred back to step 810 to get the next element in the touching list. If the answer is yes, operation is transferred to step 820.
- 10 At step 820, the merged dynamic borders are rendered and then at step 830 the core and dynamic border is displayed. When the user moves an element 10 toward another element 10a, a determination is made if the borders of elements 10 and 10a are going to overlap and this process is done for all additional elements that overlap or touch element 10 as shown in Figure 8.
- 15 When two elements merge, the merged portion is calculated using the addition of two height arrays as shown graphically in Figure 6.

When an edge 14 merges with more than one other edge, the arrays corresponding to the touching edges 14 and 14a are added to obtain the global

- 9 -

array for the merged portion which will be the merged dynamic border. The merged edge array is equal to the sum of all the edge arrays of the elements in the merged group.

The melted color aspect of the edges is achieved by using the height 5 values of each edge array to mix the colors of each element doing a weighted average. For two colors, say C1 and C2, and using the height of each edge, say a1 and a2, the melted or final color, Cf, of the global edge is preferably calculated using the equation:

$$Cf = (C1 * a1 + C2 * a2) / (a1 + a2)$$

10        Creation of Groups

If the edges 14 and 14a of two or more elements 10 and 10a are merged, it is possible to freeze or "stick together" all the elements as shown in Figure 2. This arrangement will behave in the same way that a single element 10 behaves and is called a group 30. This group 30 comprises as many cores 12 as there 15 were in all the elements comprising the group 30 and the position of one of the cores 12 is fixed relative to the other cores in the group 30 as shown in Figure 2. This group 30 could be merged again with another element 10b. A group 30 behaves like a single element 10, so it is possible to create a new group 30a

- 10 -

from a set of elements composed of groups. As shown in Figure 2, a group 30a of three cores 12 is created and can be made of a single element which is yellow and another group 30 made up of a green element 10 and a red element 10a.

This grouping process can be repeated as long as elements remain to be 5 "stuck" together. This grouping procedure can be represented as a tree structure as shown in Figure 3, whose "leaves" would be simple elements (10, 10a, 10b, . . . ) and whose nodes would be the grouped elements. Once the group 30a is created, the elements of the group 30a can be removed from the group 30a. For example, as graphically illustrated in Figure 3, the elements can 10 be removed by going along the tree from the group 30a and sequentially removing the elements from the group 30a so that only single elements remain. The user configured groups can include many different designs as illustrated in Figures 9, 10 and 11.

#### Static/Dynamic Aspect of the Edge

15 In another preferred embodiment, an edge 14 can be static meaning it cannot be merged with other edges 14a. In this embodiment, the static edge 14 behaves like a standard window and passes on top of the other elements as shown in Figure 4. In one embodiment, the static edge 14 can be altered to be

a dynamic edge. For example, if an element 10 with a static edge 14 is on top of another element 10a with a dynamic edge as shown in Figure 4, the static edge 14 can be turned into a dynamic edge if the core 12 of elements 10 does not overlap with any of the cores of element 10a. For the example shown in Figure 4, the static edge 14 cannot be turned into a dynamic edge since its core 12 overlaps the cores of the overlapped or subjacent elements.

While embodiments of the present invention and modifications thereto have been shown and disclosed in the drawings and specification, alternate embodiments of the present invention will be apparent to a person of ordinary skill in the art and this application is intended to include those embodiments within the full breadth and scope of the claims. The present invention is not limited by any parameters described herein and the present invention need not include all of the features disclosed in the single embodiment, but rather one or more features may be included.